

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of

Tomoki AYABE

Serial No. (unknown)

Filed herewith

METHOD AND APPARATUS FOR
ENCODING IMAGE DATA IN
CONFORMITY WITH JOINT
BI-LEVEL IMAGE GROUP SYSTEM



CLAIM FOR FOREIGN PRIORITY UNDER 35 U.S.C. 119
AND SUBMISSION OF PRIORITY DOCUMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Attached hereto is a certified copy of applicant's corresponding patent application filed in Japan on February 23, 2000, under No. 2000-046462.

Applicant herewith claims the benefit of the priority filing date of the above-identified application for the above-entitled U.S. application under the provisions of 35 U.S.C. 119.

Respectfully submitted,

YOUNG & THOMPSON

By

Benoît Castel

Benoît Castel
Attorney for Applicant
Customer No. 000466
Registration No. 35,041
745 South 23rd Street
Arlington, VA 22202
Telephone: 703/521-2297

February 21, 2001

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

2000年 2月23日

出 願 番 号
Application Number:

特願2000-046462

出 願 人
Applicant (s):

日本電気アイシーマイコンシステム株式会社

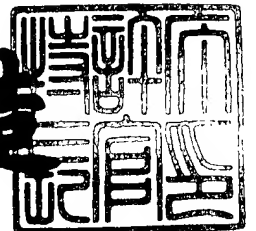


CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年12月15日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 01210911

【提出日】 平成12年 2月23日

【あて先】 特許庁長官殿

【国際特許分類】 H04N 1/41

【発明者】

【住所又は居所】 神奈川県川崎市中原区小杉町一丁目4 0 3 番 5 3 日本
電気アイシーマイコンシステム株式会社内

【氏名】 綾部 友喜

【特許出願人】

【識別番号】 000232036

【氏名又は名称】 日本電気アイシーマイコンシステム株式会社

【代理人】

【識別番号】 100065385

【弁理士】

【氏名又は名称】 山下 穰平

【電話番号】 03-3431-1831

【手数料の表示】

【予納台帳番号】 010700

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9002303

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像データ圧縮方法と J B I G 方式符号化処理方法とその装置

【特許請求の範囲】

【請求項 1】 入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2 値画素データに圧縮を行う画像データ圧縮方法において、

前記コンテキスト作成処理は、典型的予測の有無を判断し、前記典型的予測のない場合に、まず、1 ライン分が白ラインか否かを検索し、続いて 3 ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記 3 ラインオール白用のライン専用の第 1 の 1 ライン符号化処理を実行し、

前記 3 ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第 2 の 1 ライン符号化処理を実行することを特徴とする画像データ圧縮方法。

【請求項 2】 請求項 1 に記載の画像データ圧縮方法において、

前記 3 ラインオール白用のライン専用の前記第 1 の 1 ライン符号化処理は、前記符号化対象画素の周辺画素を 1 次元化したコンテキストをインデックスとして、前記予測値と状態値からなる学習テーブルにより前記状態値等を抽出し、前記状態値をインデックスとして確率推定テーブルにより予測外れの領域幅を抽出し、前記 3 ライン分の画像データの作成処理、ブロック判別処理、予測値判別処理、の通常処理を行わないことを特徴とする画像データ圧縮方法。

【請求項 3】 入力された画像データをライン毎に読み出し、その 2 つ前のラインと、1 つ前のラインと、現行ラインの所定の画像データ毎にコンテキストを作成して、符号化対象画素を 2 値画素データに変換する J B I G 方式符号化処理方法において、

前記コンテキスト作成処理は、前記 2 つ前のラインと、前記 1 つ前のラインと、現行ラインのコンテキスト対象の画素データがオール白かオール白以外かを判別し、これがオール白の場合、コンテキスト作成対象データとせず、且つ 1 ライ

ンの画像データがオール白かどうかを検索し、3ライン分の検索結果をもとに、3ラインがオール白か又はオール白でないかを判別し、オール白の場合には前記3ラインのオール白用の第1の1ライン符号化処理を施し、前記3ラインがオール白ではない場合には第2の1ライン符号化処理を実行することを特徴とするJ B I G方式符号化処理方法。

【請求項4】 請求項3に記載のJ B I G方式符号化処理方法において、上記検索結果に基いて、3ラインがオール白の場合は、前記符号化対象画素の予測値が白であることを判別し、3ラインオール白用の前記第1の1ライン符号化処理を施し、前記オール白ライン専用の1ライン符号化処理によって前記3ライン分の画像データの作成処理、ブロック判別処理と、予測値判別処理、の各通常処理を行わないことを特徴とするJ B I G方式符号化処理方法。

【請求項5】 入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎にコンテキストを作成して、前記コンテキストを参照しつつ符号化対象画素を2値画素データに変換するJ B I G方式符号化処理装置において、

前記ブロック単位の画像データがオール白かオール白以外かを判別する処理手段と、前記画像データがオール白の場合、コンテキスト作成対象データとせず且つ1ラインの画像データがオール白かどうかを検索する処理手段と、前記画像データを3ライン分検索し、その結果をもとに3ラインがオール白か又はオール白でないかを判別する処理手段と、前記3ラインがオール白の場合には3ラインオール白用の1ライン符号化処理を施す処理手段と、前記3ラインがオール白ではない場合には1ライン符号化処理を実行する処理手段とを具備することを特徴とするJ B I G方式符号化処理装置。

【請求項6】 入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2値画素データに圧縮を行う画像データ圧縮方法において、

前記コンテキスト作成処理は、典型的予測のない場合に、まず、1ライン分が白ラインか否かを検索し、続いてn（nは2以上の整数）ラインが全て白ライン

か否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記 n ラインオール白用のライン専用の第 1 の 1 ライン符号化処理を実行し、

前記 n ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第 2 の 1 ライン符号化処理を実行することを特徴とする画像データ圧縮方法。

【請求項 7】 入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2 値画素データに圧縮を行う画像データ圧縮方法をプログラムとして格納したコンピュータが読み取り可能な記録媒体において、

前記コンテキスト作成処理は、典型的予測の有無を判断し、前記典型的予測のない場合に、まず、1 ライン分が白ラインか否かを検索し、続いて 3 ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記 3 ラインオール白用のライン専用の第 1 の 1 ライン符号化処理を実行し、前記 3 ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第 2 の 1 ライン符号化処理を実行することを特徴とする画像データ圧縮方法を前記プログラムとして格納したコンピュータが読み取り可能な記録媒体。

【請求項 8】 入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成し、2 値画素データに圧縮を行う画像データ圧縮方法をプログラムとして格納したコンピュータが読み取り可能な記録媒体において、

前記コンテキスト作成処理は、典型的予測のない場合に、まず、1 ライン分が白ラインか否かを検索し、続いて n (n は 2 以上の整数) ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記 n ラインオール白用のライン専用の第 1 の 1 ライン符号化処理を実行し、前記 n ラインが全て白ライ

ンでない場合、及び前記コンテキストの予測値が白でない場合に第2の1ライン符号化処理を実行することを特徴とする画像データ圧縮方法を前記プログラムとして格納したコンピュータが読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、画像データ圧縮方法とJBIG方式符号化処理方法とその装置及び該JBIG方式符号化処理方法をプログラムとして格納したコンピュータが読み取り可能な記録媒体に関する。

【0002】

【従来の技術】

画像データの符号化方式として、ソフトコピー通信に適したプログレッシブ符号化方式の一つとして、国際標準ITU-Tの1995年8月勧告T. 85によるJBIG (Joint Bi-label Image Group) 符号化方式（ファクシミリ版）が勧告されている。

【0003】

従来のファクシミリ方式のMH (Modified Huffman: データの圧縮・伸張技法) /MR (Modified READ: データの圧縮・伸張技法) /MMR (Modified Modified READ: データの圧縮・伸張技法) では、符号化処理時に画素データから白あるいは黒の色の連続する画素の長さであるランレングス値を求め、このランレングスに対応した符号を符号化テーブルより求めている。

【0004】

これに対し、JBIG方式では、符号化対象画素1画素の圧縮処理において、符号化対象画素の周辺10画素を参照し、符号化対象画素が白か黒かを予測し、予測が実際に外れた場合のみ、算術符号化方式により符号化を行っている。

【0005】

ITU-Tの勧告T. 85によるJBIG (Joint Bi-label Image Group) 方式の従来のデータ符号化方法について図面を参照しながら説明する。この従来技術について、特開平9-149264号公報（特許2793536号）には、以

下のことが記載されている。

【0006】

図7に示すように、この装置は、演算処理、データ処理、および各部の制御処理等を行うCPU（中央処理演算装置）100と、送信時に読み取った原稿を2値画像データにしたものを蓄える画像データ・メモリ400と、JBIGによる符号化プログラム201および画像データを符号データに変換するための確率推定テーブル・メモリ202が配置格納されるROM200と、符号化対象画素が白か黒かを予測して当たる確率を高くするために参照される学習テーブル・メモリ300と、変換された結果である符号データをFIFO（ファーストインファーストアウト）管理するFIFOメモリ500とがバス600で接続されている。

【0007】

また、CPU100は、符号化対象ラインよりブロック単位で読み出した画像データを保持するレジスタ101と、符号化対象ラインの1つ前のラインよりブロック単位で読み出した画像データを保持するレジスタ102と、符号化対象ラインの2つ前のラインよりブロック単位で読み出した画像データを保持するレジスタ103と、予測結果を保持するレジスタ104と、学習テーブル・メモリ300より読み出した予測値、状態値を保持するレジスタ105と、確率推定テーブル・メモリ202より読み出した外れの領域幅を保持するレジスタ106と、画素の前後関係を示すコンテキストの内容を保持するレジスタ107とを有している。

【0008】

前記CPU100は、以降に述べるJBIG方式による符号化プログラム201を実行することによって、データ符号化方法を実現する。

【0009】

全体の動作について、図8のフローチャートを参照しながら説明する。まず、典型的予測をするかしないか判別し（ステップS21）、典型的予測をしない場合は、1ライン符号化処理を行う（ステップS26）。次に原稿ライン数分の処理が終了したか否か判別し（ステップS27）、終了していない場合は、1ライ

ン符号化処理（ステップS 2 6）へ戻り、符号化処理を繰り返す。1 ページ分が終了した場合に処理を終了する。

【0 0 1 0】

前記典型的予測をするかしないかの判別により（ステップS 2 1）、典型的予測をする場合は、符号化対象ラインと符号化対象ラインの1つ前のラインとを比較して（ステップS 2 9）、2つのラインが同じかどうかを判別する（ステップS 2 1 0）。前記判別結果が同じでなかった場合は、1ライン符号化処理（ステップS 2 6）を実行し、2つのラインが同じである場合と同様に、1 ページ分が終了したか否かを判別し（ステップS 2 7）、終了していない場合は、ライン比較処理（ステップS 2 9）へ戻り、符号化処理を繰り返す。以上の処理を原稿ライン数分繰り返すことで符号化処理を行う。

【0 0 1 1】

次に、1ラインの符号化処理（ステップS 2 6）について、図9を参照しながら説明する。まず、ラインが終了したか否かの判別を行う（ステップS 3 1）。終了していれば処理を終了する。終了していなければ、符号化対象ラインの2つ前のラインH 2、符号化対象ラインの1つ前のラインH 1、符号化対象ラインP I Xのブロック単位の画像データを作成する（ステップS 9 0、S 9 1、S 9 2）。具体的には、指定アドレスからの画像データの読み出しであり、この時、コンテキストの構造上、符号化対象画素以前のデータが必要になるため、前記読み出した画像データをシフトして、前回のブロック処理の残りの画像データと結合することで、ブロックの画像データとしている。

【0 0 1 2】

次に、前記作成した3つの画像データ・ブロックがオール白かどうかを判別する（ステップS 9 3）。前記判別結果がオール白の場合は、ブロック処理が終了するまで、1画素符号化処理（ステップS 9 5）を繰り返す。

【0 0 1 3】

前記ステップS 9 3での判別結果がオール白でない場合は、ブロック処理が終了するまで（ステップS 4 6）、符号化対象画素の周辺画素を1次元化したコンテキストの作成（ステップS 9 4）、1画素符号化処理（ステップS 9 5）、3

ブロック分の画像データをそれぞれ1ビット・シフト（ステップS96）する一連の処理を繰り返す。

【0014】

ブロック単位分の処理が終了した場合は、1ライン符号化処理の先頭に戻り、1ライン分の符号化処理が終了したかどうかを判別する（ステップS31）。

【0015】

以上の処理を原稿ライン数分繰り返すことで符号化処理を行う。

【0016】

次に、前記1画素符号化処理の手順を、図10から図12を参照しながら説明する。まず、符号化対象画素の周辺画素を1次元化した画素の前後関係を示すコンテキストをインデックスとして、学習テーブル300より予測値、状態値を読み出す（ステップS100）。次に、前記状態値をインデックスとして、確率推定テーブル202より外れの領域幅を読み出す（ステップS41）。

【0017】

さらに、白黒の順列が現われる確率を示す領域幅から前記外れの領域幅を差し引いて、新しい領域幅に更新する（ステップS42）。ここで、符号化対象画素が予測値と一致しているかを判別する（ステップS110）。

【0018】

前記判別結果が予測外れの場合は、外れの確率を示す領域幅を変更し、更新した状態値を学習テーブル300に書き込む予測外れ処理を実行し（ステップS120）、領域幅が小さくなり、領域幅を外れの領域幅より大きくする正規化を行う（ステップS45）。一方、符号化対象画素が予測値と一致している場合（ステップS110）、即ち前記判別結果が予測当たりの場合は、領域幅が小さくなり領域幅を外れの領域幅より大きくする正規化が必要かどうかを判別する（ステップS43）。前記判別結果で正規化が必要な場合は、当たりの確率を示す領域幅を変更し、更新した状態値を学習テーブル300に書き込む予測当り処理を行い（ステップS44）、続いて正規化処理を行う（ステップS45）。

【0019】

また、図11（a）は、図7の画像データメモリ400のメモリブロックの概

要を示し、書類の1ページを模して、横軸に各ラインの符号化方向、縦軸に垂直方向の状態を示している。特にJBI G方式による予測符号化のため、符号化対象ライン(PIX)と、その前の符号化対象ラインの一つ前のライン(H1)と、その前の符号化対象ラインの二つ前のライン(H2)のデータ状態を示している。図11(b)は、図11(a)の3つのラインについて、それぞれ各ラインを画素に分けたメモリ・ブロックの詳細を示している。

【0020】

符号化対象画素を予測するために、図11(b)の斜線部に示すように、2つ前のラインH2から3画素、一つ前のラインH1から5画素、現行のPIXラインから2画素が予測対象の周辺画素(モデルテンプレート)となる。その予測用の10画素から、図11(c)に示すように、順番符号0~9を付けて、図11(d)に示すような、0~9画素からなるコンテキストを作成する。このコンテキストを用いて符号化対象画素が、JBI G方式の規格に則って、白か黒かの予測を立てる。

【0021】

この予測の当たり・外れを検出し、外れの場合に、算術符号化により圧縮を行う。同時にコンテキストの値毎に予測の当たり外れを判断し、図7に示した学習テーブル300に保持する。この学習テーブル300を用いることで、予測が当たる確率を高め、予測確率を高めることによって、圧縮符号データを小さくすることができる。

【0022】

図12は、コンテキストテーブルともいわれる学習テーブル300と、確率推定テーブル202の例である。学習テーブル300には、モデルテンプレート(10画素)を一次元化したコンテキスト毎に、予測値と状態値とがテーブル形式に格納されており、確率推定テーブル202には、その予測値から、予測外れの領域幅と、予測外れ時の状態値と、予測当たり時の状態値と、予測外れ時の条件設定とが、統計的手法によって求められ、テーブル形式に格納されている。

【0023】

上記説明したように、この従来技術では、JBI G方式を参考として、符号化

対象画素の周辺画素を参照する2値画像データ圧縮方法を説明している。また、符号化対象ラインと、符号化対象ラインの1ライン前のラインを比較して、同じであればこの符号化対象ラインを符号化対象から除外しているので、コンテキスト作成処理回数の削減と、画像メモリの読み出し回数を削減して、コンテキスト作成回数を減らして、高速化している。

【0024】

上記説明したように、JBIG方式を用いた従来技術の画像圧縮方法では、コンテキストから符号化対象画素の白と黒とを予測して、予測が違う場合だけを符号化する。また、符号データの量を減少させるため、その予測を学習して予測はずれを少なくしている。こうして、従来のMH/MR/MMR方式よりも、文字中心の文書で1.1～1.5倍の高い圧縮性能が得られている。

【0025】

【発明が解決しようとする課題】

しかし、この従来技術には、次のような問題点があった。

第1の問題点は、上記従来技術によるJBIG方式のオプションである典型的予測をしない場合、符号化処理速度が遅いということである。その理由は、コンテキスト作成のための画像データ・ブロックの作成処理が、符号化対象ライン、1つ前のライン、2つ前のラインの3つのラインに対して、ブロック単位で頻繁に行われるからである。また、もう一つの理由は、コンテキストの構造上、符号化対象画素以前のデータが必要で、符号化対象画素の画像データをロードした後に、以前のデータと結合して画像データ・ブロックを作成する必要があるからである。

【0026】

このため、符号化対象画素からの次の画像データが、レジスタ幅全体へロードできず、レジスタ幅より小さい幅で画像データをロードしている。これが、データのロード回数を増やす原因となっている。

【0027】

そこで、本発明は、従来の画像データ圧縮方法或いはJBIG方式を用いた符号化処理方法において、プログラム処理速度を高速化することを課題とする。さ

らに、当該高速化した J B I G 方式を用いた符号化処理方法を実行するプログラムを格納した記録媒体を提供することを課題とする。

【 0 0 2 8 】

【課題を解決するための手段】

本発明は、上記課題を解決するために、入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2 値画素データに圧縮を行う画像データ圧縮方法において、前記コンテキスト作成処理は、典型的予測の有無を判断し、前記典型的予測のない場合に、まず、1 ライン分が白ラインか否かを検索し、続いて3 ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記3 ラインオール白用のライン専用の第1の1 ライン符号化処理を実行し、前記3 ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第2の1 ライン符号化処理を実行することを特徴とする。

【 0 0 2 9 】

また、本発明は、入力された画像データをライン毎に読み出し、その2 つ前のラインと、1 つ前のラインと、現行ラインの所定の画像データ毎にコンテキストを作成して、符号化対象画素を2 値画素データに変換する J B I G 方式符号化処理方法において、

前記コンテキスト作成処理は、前記2 つ前のラインと、前記1 つ前のラインと、現行ラインのコンテキスト対象の画素データがオール白かオール白以外かを判別し、これがオール白の場合、コンテキスト作成対象データとせず、且つ1 ラインの画像データがオール白かどうかを検索し、3 ライン分の検索結果をもとに、3 ラインがオール白か又はオール白でないかを判別し、オール白の場合には前記3 ラインのオール白用の第1の1 ライン符号化処理を施し、前記3 ラインがオール白ではない場合には第2の1 ライン符号化処理を実行することを特徴とする。また、本発明は、入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎にコンテキストを作成して、前記コンテキスト

を参照しつつ符号化対象画素を2値画素データに変換するJ B I G方式符号化処理装置において、前記ブロック単位の画像データがオール白かオール白以外かを判別する処理手段と、前記画像データがオール白の場合、コンテキスト作成対象データとせず且つ1ラインの画像データがオール白かどうかを検索する処理手段と、前記画像データを3ライン分検索し、その結果をもとに、3ラインがオール白か又はオール白でないかを判別する処理手段と、前記3ラインがオール白の場合には3ラインオール白用の1ライン符号化処理を施す処理手段と、前記3ラインがオール白ではない場合には1ライン符号化処理を実行する処理手段とを具備することを特徴とする。

【 0 0 3 0 】

また、本発明は、入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2値画素データに圧縮を行う画像データ圧縮方法において、前記コンテキスト作成処理は、典型的予測のない場合に、まず、1ライン分が白ラインか否かを検索し、続いて n (n は2以上の整数)ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記 n ラインオール白用のライン専用の第1の1ライン符号化処理を実行し、前記 n ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第2の1ライン符号化処理を実行することを特徴とする。

【 0 0 3 1 】

また、本発明は、入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2値画素データに圧縮を行う画像データ圧縮方法をプログラムとして格納したコンピュータが読み取り可能な記録媒体において、前記コンテキスト作成処理は、典型的予測の有無を判断し、前記典型的予測のない場合に、まず、1ライン分が白ラインか否かを検索し、続いて3ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記3

ラインオール白用のライン専用の第1の1ライン符号化処理を実行し、前記3ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第2の1ライン符号化処理を実行することを特徴とする。

【0032】

また、本発明は、入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成し、2値画素データに圧縮を行う画像データ圧縮方法をプログラムとして格納したコンピュータが読み取り可能な記録媒体において、前記コンテキスト作成処理は、典型的予測のない場合に、まず、1ライン分が白ラインか否かを検索し、続いてn（nは2以上の整数）ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記nラインオール白用のライン専用の第1の1ライン符号化処理を実行し、前記nラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第2の1ライン符号化処理を実行することを特徴とする。

【0033】

また、本発明は、図面を参照して説明すれば、JBIGによる符号化処理を高速化するために、図2において、典型的予測をしない場合に、符号化対象ラインの画像データが1ラインすべて白画素かどうかを判別する1ライン検索処理（ステップS22）と、符号化対象ラインから2ライン前までの3ライン分の検索結果をもとに、3ラインがオール白か、オール白でないかを判別する処理（ステップS23）と、前記検索結果がオール白の場合に、予測値が白であるかを判別する処理（ステップS24）により、3ライン・オール白専用の1ライン符号化処理（ステップS25）へ分岐させることで、3ライン分の画像データの作成処理、ブロック判別処理、予測値判別処理、を削減し、符号化処理の高速化を図る。

【0034】

【発明の実施の形態】

本発明による実施形態について、図面を参照しつつ詳細に説明する。

【0035】

【第一の実施形態】

(1) 構成の説明

以下、本発明の第一の実施形態について図面を参照しながら説明する。図1は本発明の第一の実施形態を説明するブロック図である。

【0036】

図1に示すように、本第一の実施形態は、J B I G方式符号化処理装置700の内部に、演算処理、データ処理、および各部の制御処理等を行うCPU（中央処理演算装置）100と、不図示の読み取りセンサーで原稿から読み取った画像信号を2値画像データにしたものを蓄える画像データ・メモリ400と、J B I G方式による符号化するためにCPU100が動作するプログラム201Aおよび画像データを符号データに変換するための確率推定テーブル202とが配置・格納されるROM200と、符号化対象画素が白か黒かを予測して当たる確率を高くするために参照される学習テーブル・メモリ300と、F I F O（ファースト・イン ファースト・アウト）500とが、P C Iバスや、A G Pバス、Cardバス等のバスで接続されている。なお、F I F O500は先入れ先出しのメモリである。

【0037】

また、RAM300～500は、格納領域の区別が付けば、同一パッケージ内に一緒に組み合わせてもよい。また、記録媒体ドライバ650は、本ROM200内のJ B I Gプログラム201Aや確率推定テーブル202、学習テーブル300等のデータを模写して、フロッピーディスク等のパッケージメディア用のデータを、格納するものである。

【0038】

また、ROM200に格納されたJ B I Gプログラムに従って動作するCPU100は、符号化対象ラインよりブロック単位で読み出した画像データを保持するレジスタ101と、符号化対象ラインの1つ前のラインよりブロック単位で読み出した画像データを保持するレジスタ102と、符号化対象ラインの2つ前のラインよりブロック単位で読み出した画像データを保持するレジスタ103と、予測結果を保持するレジスタ104と、学習テーブル・メモリ300より読み出

した予測値、状態値を保持するレジスタ105と、確率推定テーブル・メモリ202より読み出した外れの領域幅を保持するレジスタ106とからなる。

【0039】

ここで、ブロック単位とは、画像データメモリ400の各ラインから例えば16画素或いは8画素を1ブロックとして読み出す単位である。

【0040】

さらに、コンテキストの内容を保持するレジスタ107と、符号化対象ラインがオール白かどうかを検索した結果、符号化対象ラインの1つ前のラインがオール白かどうかを検索した結果および符号化対象ラインの2つ前のラインがオール白かどうかを検索した結果のフラグを保持するレジスタ108との各レジスタを保有し、JBIGプログラム201Aに応じて各レジスタに必要なデータが格納される。

【0041】

前記CPU100は、以降に述べるJBIG方式による符号化プログラム201Aを実行することによって、本発明による第一の実施形態を実現する。

【0042】

また、ROM200にはJBIGプログラム201A及び確率推定テーブル202を固定的に格納している例を示したが、当該プログラムを変更しようとする場合には、フラッシュメモリ、EEPROM等のメモリを用いてもよいし、JBIGプログラムを外部記録媒体から読み込んで、DRAMやSRAM等に読み込んで使用してもよい。

【0043】

また、RAM300に書き込まれた学習テーブルは、本JBIGプログラムが、繰り返しの動作中に、学習した結果をテーブル状態に格納したものである。

【0044】

また、FIFOメモリ500は、画像データを符号データとして変換したデータを格納したもので、高速動作が可能であれば、SRAMやDRAMであってもよく、動作的には画像データを画像データメモリから読み込んだ後、CPU100でJBIG圧縮し、FIFOメモリ500に書き込むことになる。

【 0 0 4 5 】

また、記録媒体ドライバ 6 5 0 は、J B I G 方式の J B I G プログラム 2 0 1 A の内容をインストールするための記録媒体をドライブする。また、J B I G プログラム 2 0 1 A や確率推定テーブル等のデータを複写して記録媒体に格納する記録媒体用ドライバである。さらに、確率推定テーブル 2 0 2 のテーブル内容、学習テーブルメモリ 3 0 0 の学習テーブル内容を複写する記録媒体用ドライバとしてもよい。逆に、記録媒体ドライバ 6 5 0 に、J B I G プログラム 2 0 1 A 、確率推定テーブル 2 0 2 、学習テーブルメモリ 3 0 0 の各内容を記録した記録媒体を挿入して、C P U 1 0 0 の読み出し動作で、複写することによって、J B I G 方式の動作を実行させることができる。

【 0 0 4 6 】

J B I G 方式符号化処理装置 7 0 0 は、上述の C P U 1 0 0 と R O M 2 0 0 , R A M 3 0 0 , R A M 4 0 0 , R A M 5 0 0 , 記録媒体ドライバ 6 5 0 とを備え、J B I G 方式による圧縮符号化処理を実行する装置である。

【 0 0 4 7 】

画像メモリ 8 0 0 は、スキャナやデジタルカメラ、光電変換装置等から画像データを入力して、一時的に記録する記録媒体で、原則的に、図 1 1 (a) に示すように、複数ラインのライン毎に画像データが書き込まれる。

【 0 0 4 8 】

また、伝送系 9 0 0 は、J B I G 方式符号化処理装置 7 0 0 によって符号化された画像データを外部に、例えばファクシミリや、インターネットを介して他のパーソナルコンピュータ等に出力する。また、この画像データをフロッピーディスクや M O ディスク等に出力することもできる。

【 0 0 4 9 】

(2) 動作の説明

次に、第一の実施形態の J B I G 方式による符号化プログラム 2 0 1 A の手順を、図 2 から図 4 のフローチャートを参照しながら説明する。この J B I G 方式による画像データの符号化は、スキャナーや光電変換装置、ビデオカメラ等によって、対象原稿や対象画像を読み取って、ファクシミリ用やイーメール用にその

画像データを本方式による符号データ変換して、外部に出力するために用いられる。

【 0 0 5 0 】

概略的に本実施形態について説明すれば、図 1 及び図 1 1、図 1 2 を参照して、入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2 値画素データに圧縮を行う画像データ圧縮方法であり、コンテキスト作成処理は、典型的予測の有無を判断し、典型的予測のない場合に、まず、1 ライン分が白ラインか否かを検索し、続いて 3 ラインが全て白ラインか否かを検索・判断し、オール白の場合コンテキストの予測値が白であるのか否かを判断し、コンテキストの予測値が白である場合に 3 ラインオール白用のライン専用の第 1 の 1 ライン符号化処理を実行し、3 ラインが全て白ラインでない場合、及びコンテキストの予測値が白でない場合に、第 2 の 1 ライン符号化処理を実行する。

【 0 0 5 1 】

また、上記検索結果において、3 ラインがオール白の場合は、コンテキストを作成せず処理を削減している。また、通常の処理として、画像データの圧縮のため、学習テーブル 3 0 0 から予測値から状態値を読み出し、その状態値から確率推定テーブル 2 0 2 によって、外れの領域幅、予測外れ時の状態値、予測当たり時の状態値等を読み出し、J B I G 方式の符号化処理を行うことで、画像データを符号化する。

【 0 0 5 2 】

また、3 ラインがオール白の場合に、オール白ライン専用の 1 ライン符号化の際、3 ライン分の画像データの作成処理、ブロック判別処理、予測値判別処理、を行わないので、処理速度を高速化できる。

【 0 0 5 3 】

つぎに、全体の動作について、図 2 を参照しながら説明する。
まず、1 ページ分の符号化処理の先頭で、オール白ライン・フラグ (P I X : 符号化対象ライン、H 1 : 1 ライン前、H 2 : 2 ライン前の 3 個のフラグ) の初期

化を行う（ステップ S 2 0）。

【 0 0 5 4 】

これは、符号化対象ラインが、ページ先頭の場合は、2ライン前および1ライン前のラインが存在しないことになるが、実際には仮想の白ラインがあることを前提に処理を行うためである。

【 0 0 5 5 】

次に、符号化対象ライン P I X と符号化対象ラインの1つ前のライン H 1 とを比較する典型的予測をするかしないかを判別し（ステップ S 2 1）、典型的予測をしない場合、符号化対象のラインが、1ラインすべて白画素かどうかを判別する1ライン検索処理を行う（ステップ S 2 2）。前記検索処理により、符号化対象ラインの画像データがオール白の場合は、オール白ラインフラグ P I X をセット、オール白でない場合は、オール白ラインフラグ P I X をリセットする。

【 0 0 5 6 】

ここで、符号化対象ラインのオール白ラインフラグ P I X と、前2ライン分のオール白ラインフラグ H 1、H 2 により、3ラインの画像データがオール白か、オール白でないかを判別する（ステップ S 2 3）。前記判別処理により、3ラインの画像データがオール白の場合は、予測値が白であるかどうかを判別し（ステップ S 2 4）、予測値が白であれば、3ラインがすべて白画素用の1ライン符号化処理を行う（ステップ S 2 5）。前記判別結果が、オール白でない場合、あるいは前記予測値の判別で、予測値が白でない場合（黒）は、従来どおりの1ライン符号化処理を行う（ステップ S 2 6）。

【 0 0 5 7 】

次に、原稿ライン数分の処理が終了したかを判別し（ステップ S 2 7）、終了していない場合は、1ライン検索処理（ステップ S 2 2）へ戻り、符号化処理を繰り返す。

【 0 0 5 8 】

前記典型的予測をするかしないかの判別により（ステップ S 2 1）、典型的予測をする場合は、従来と同様に、符号化対象ラインと符号化対象ラインの1つ前のラインとを比較して（ステップ S 2 9）、2つのラインが同じかどうか判別す

る（ステップ S 2 1 0）。前記判別結果が同じでなかった場合は、1ライン符号化処理（ステップ S 2 6）を、同じである場合は、符号化対象ラインを符号化の対象としない。

【 0 0 5 9 】

次に、典型的予測をしない場合と同様に、原稿ライン数分の処理が終了したかを判別し（ステップ S 2 7）、終了していない場合は、ライン比較処理（ステップ S 2 9）へ戻り、符号化処理を繰り返す。

【 0 0 6 0 】

以上の処理を原稿ライン数分繰り返すことで符号化処理を行う。

次に、1ライン検索処理（ステップ S 2 2）および3ラインがオール白、予測値が白の場合の3ライン・オール白用1ライン符号化処理（ステップ S 2 5）について、図3を参照しながら説明する。

【 0 0 6 1 】

図3（a）において、1ラインの検索処理では、まず、オール白ラインフラグについて、1ライン前のH1から2ライン前のH2に、符号化対象ラインのPIXから1ライン前のH1にそれぞれ代入するフラグ再設定処理（ステップ S 3 2）を行う。

【 0 0 6 2 】

次に、オール白ラインであるか否かを判別する符号化対象ラインの先頭アドレスを設定（ステップ S 3 3）後、前記アドレスから画像データを読み出す（ステップ S 3 4）。従来、画像データのブロック作成処理時の読み出しがレジスタ幅全体にできなかったが、ここでは、コンテキストの作成を行わないため、レジスタ幅全体への読み出しが可能となり、画像データのロード回数を削減することができる。

【 0 0 6 3 】

次に、前記画像データがオール白か否かを判別（ステップ S 3 5）し、オール白の場合は、アドレスを更新して次の画像データのアドレス設定を行う。

【 0 0 6 4 】

以上の処理を1ラインが終了するまで繰り返す（ステップ S 3 7）。前記検索

結果により、すべての画像データが白であった場合には、オール白ラインフラグ P I X をセット（ステップ S 3 8）して、1ライン検索処理を終了する。一方、前記検索処理（ステップ S 3 5）で、オール白以外が検出された場合は、オール白ラインフラグ P I X をリセット（ステップ S 3 9）して、1ライン検索処理を終了する。

【 0 0 6 5 】

図 3（b）において、1ラインの符号化処理は、1画素符号化処理（ステップ S 3 0）を1ライン分の処理（ステップ S 3 1）が終了するまで、繰り返し行うのみである。ここでは、従来行っていた符号化対象ライン、1ライン前と2ライン前の画像データの作成処理と、ブロック単位のオール白判別処理と、ブロック終了判別処理は行わない。これは、3ラインがオール白の場合、1ライン符号化処理中のモデルテンプレートがすべて 0（コンテキストが 0）で変更がない（不変）ためである。

【 0 0 6 6 】

次に、前記 1 画素符号化処理（ステップ S 3 0）について、図 4 を参照しながら説明する。

【 0 0 6 7 】

まず、符号化対象画素の周辺画素を 1 次元化したコンテキスト（0）をインデックスとして、学習テーブル 3 0 0 より予測値、状態値を読み出す（ステップ S 4 0）。次に、前記状態値をインデックスとして、確率推定テーブル 2 0 2 より外れの領域幅を読み出す（ステップ S 4 1）。さらに、白黒の順列が現われる確率を示す領域幅を前記外れの領域幅を差し引いて、新しい領域幅に更新する（ステップ S 4 2）。ここで、確率推定テーブル 2 0 2 は、勧告 T. 8 2 による J B I G 符号化方式の規定により定まったテーブルであり、図 1 2 を用いて説明したとおりである。

【 0 0 6 8 】

ここで、従来であれば、図 1 0 に示して説明したように、予測値の判別処理が必要であったが、本発明においては、すでに予測値が白と判っているため、判別処理が不要となる。

【 0 0 6 9 】

次に、前記領域幅の更新処理で領域幅が小さくなり、領域幅を外れの領域幅より大きくする正規化が必要かどうかを判別する（ステップ S 4 3）。前記判別結果で正規化が必要な場合は、当たりの確率を示す領域幅を変更し、更新した状態値を学習テーブル 3 0 0 に書き込む予測当たり処理（ステップ S 4 4）、正規化処理を行う（ステップ S 4 5）。

【 0 0 7 0 】

正規化が必要ない場合は、ブロックの終了判別を行い（ステップ S 4 6）、終了していない場合は、領域幅の更新処理（ステップ S 4 2）へ戻る。これは、予測値、状態値、外れの領域幅が変わらないため、読み出しの必要がないからである。

【 0 0 7 1 】

以上の処理を行うことにより、符号化速度を向上することができる。

【 0 0 7 2 】

本発明は、ファクシミリで取り扱うことが多い文書画像のように、白目勝ちの画像に対して、特に有効となることは明らかである。この J B I G 符号化された画像データは伝送系 9 0 0 を介して、対応するファクシミリに送信する。

【 0 0 7 3 】

また、上記 J B I G 方式のプログラムを記録媒体ドライバ 6 5 0 に挿入した記録媒体にコピーして、記録パッケージとして出力することもでき、当該記録パッケージを他のパーソナルコンピュータによって、J B I G 方式のアプリケーションプログラムとして、活用することができる。

【 0 0 7 4 】

〔第二の実施形態〕

次に、本発明の第二の実施形態について図面を参照しながら説明する。

【 0 0 7 5 】

上述した本発明の第一の実施形態では、モデルテンプレートが 3 ラインの場合について説明したが、第二の実施形態として、モデルテンプレートが n ラインの場合について記載する。 n は 2 以上の整数とし、 n が大きいほど予測確率が高く

なり、予測外れが小さくなる。現在、勧告 T. 85 による J B I G 符号化方式の規定によるモデルテンプレートは、2 ライン / 3 ラインの 2 つがあり、基本的には、 $n = 2$ 又は 3 であり、多様性に応じている。

【 0 0 7 6 】

図 5 を参照すると、本実施形態は、図 1 で説明した第一の実施形態の C P U レジスタが 3 ライン分（レジスタ 1 0 1 - 1 0 3）であるのに対し、 n ライン分のレジスタを有する点で異なる。また、J B I G 方式符号化処理装置 7 0 0 と、画像メモリ 8 0 0 と、伝送系 9 0 0 とは図示していないが、これらのブロックを含んでいるものとする。

【 0 0 7 7 】

次に、全体の動作について、図 6 を参照しながら説明する。

【 0 0 7 8 】

本実施形態は、図 2 で説明した第一の実施形態のオール白ライン・フラグの数が、3 個（3 ライン分）であるのに対し、 n 個（ n は 2 以上のライン分）分有する点で異なる。

【 0 0 7 9 】

まず、1 ページ分の符号化処理の先頭で、オール白ライン・フラグ（P I X : 符号化対象ライン、 $H(n-1) : (n-1)$ ライン前 . . . の n 個のフラグ）の初期化を行う（ステップ S 6 0）。

【 0 0 8 0 】

次に、符号化対象ラインと符号化対象ラインの 1 つ前のラインとを比較する典型的予測をするかしないかを判別し（ステップ S 2 1）、典型的予測をしない場合は、符号化対象のラインが、1 ラインすべて白画素かどうかを判別する 1 ライン検索処理を行う（ステップ S 2 2）。前記検索処理により、符号化対象ラインの画像データがオール白の場合は、オール白ラインフラグ P I X をセット、オール白でない場合は、オール白ラインフラグ P I X をリセットする。

【 0 0 8 1 】

ここで、オール白ラインフラグ P I X と、前（ $n-1$ ）ライン分のオール白ラインフラグにより、 n ラインの画像データがオール白か、オール白でないかを判

別する（ステップ S 6 2）。前記判別処理により、n ラインの画像データがオール白の場合は、コンテキストの予測値が白であるかどうかを判別し（ステップ S 2 4）、予測値が白であれば、n ラインがすべて白画素用の 1 ライン符号化処理を行う（ステップ S 6 3）。前記判別結果が、オール白でない場合、あるいは前記予測値の判別で、予測値が白でない場合（黒）は、従来どおりの 1 ライン符号化処理を行う（ステップ S 2 6）。

【 0 0 8 2 】

次に、原稿ライン数分の処理が終了したかを判別し（ステップ S 2 7）、終了していない場合は、1 ライン検索処理（ステップ S 2 2）へ戻り、符号化処理を繰り返す。

【 0 0 8 3 】

次に、典型的予測をするかしないかの判別により（ステップ S 2 1）、典型的予測をする場合は、従来と同様に、符号化対象ラインと符号化対象ラインの 1 つ前のラインとを比較して（ステップ S 2 9）、2 つのラインが同じかどうか判別する（ステップ S 2 1 0）。

【 0 0 8 4 】

前記判別結果が同じでなかった場合は、1 ライン符号化処理（ステップ S 2 6）を、同じである場合は、符号化対象ラインを符号化の対象としない。

【 0 0 8 5 】

次に、典型的予測をしない場合と同様に、原稿ライン数分の処理が終了したかを判別し（ステップ S 2 7）、終了していない場合は、ライン比較処理（ステップ S 2 9）へ戻り、符号化処理を繰り返す。以上の処理を原稿ライン数分、繰り返すことで符号化処理を行う。

【 0 0 8 6 】

次に、1 ライン検索処理（ステップ S 2 2）について、図 3（a）を参照しながら説明する。基本的には、第一の実施形態で説明した 3 ラインの場合と同様であるが、フラグ再設定処理（ステップ S 3 2）におけるオール白ラインフラグの代入処理が n ライン分行われる点で異なる。

【 0 0 8 7 】

次に、nラインがオール白、予測値が白の場合のnライン・オール白用1ライン符号化処理（ステップS63）について、図3（b）を参照しながら説明する。

【0088】

1ラインの符号化処理は、1画素符号化処理（ステップS30）を1ライン分の処理（ステップS31）が終了するまで、繰り返し行うのみである。

【0089】

以降の処理（1画素符号化処理）については、第一の実施形態と同様である。

【0090】

以上の処理を行うことにより、第一の実施形態と同様に、符号化速度を向上することができる。

【0091】

【発明の効果】

本発明の画像データ圧縮方法及びJBIG方式符号化処理方法によれば、1ラインの画像データがオール白かどうかを検索し、3ライン分の検索結果をもとに、3ラインがオール白か、オール白でないかを判別する処理と、前記判別結果がオール白の場合に、予測値が白であるかを判別する処理により、3ライン・オール白専用の1ライン符号化処理へ分岐させることで、3ライン分の画像データの作成処理、ブロック判別処理、予測値判別処理を削減できるので、符号化速度を向上（従来と比べ、数十%の高速化が見込める）することができる。

【0092】

また、従来、コンテキストの構造上、画像データがレジスタ幅全体へロードすることができなかったが、1ラインのオール白検索時にレジスタ幅全体へロードすることで、画像データのロード回数を削減できるので、符号化速度を向上することができ、消費電力の点でも、大幅に削減することができる。

【図面の簡単な説明】

【図1】

本発明による第一の実施形態のJBIG方式符号化処理装置のブロック図である。

【図 2】

本発明による第一の実施形態の符号化手順を示すフローチャートである。

【図 3】

本発明による図 2 のステップ S 2 2、ステップ S 2 6 処理の詳細な手順を示すフローチャートである。

【図 4】

本発明による図 3 のステップ S 3 0 処理の詳細な手順を示すフローチャートである。

【図 5】

本発明による第二の実施形態の J B I G 方式符号化処理方法のブロック図である。

【図 6】

本発明による第二の実施形態の符号化手順を示すフローチャートである。

【図 7】

従来例の J B I G 方式符号化処理装置のブロック図である。

【図 8】

従来例の符号化手順を示すフローチャートである。

【図 9】

図 8 のステップ S 2 6 処理の詳細な手順を示すフローチャートである。

【図 1 0】

図 9 のステップ S 9 5 処理の詳細な手順を示すフローチャートである。

【図 1 1】

画像データメモリのメモリブロック概要と、ライン構造と、モデルテンプレートとコンテキストの構成図である。

【図 1 2】

学習テーブル、確率推定テーブルの構成図である。

【符号の説明】

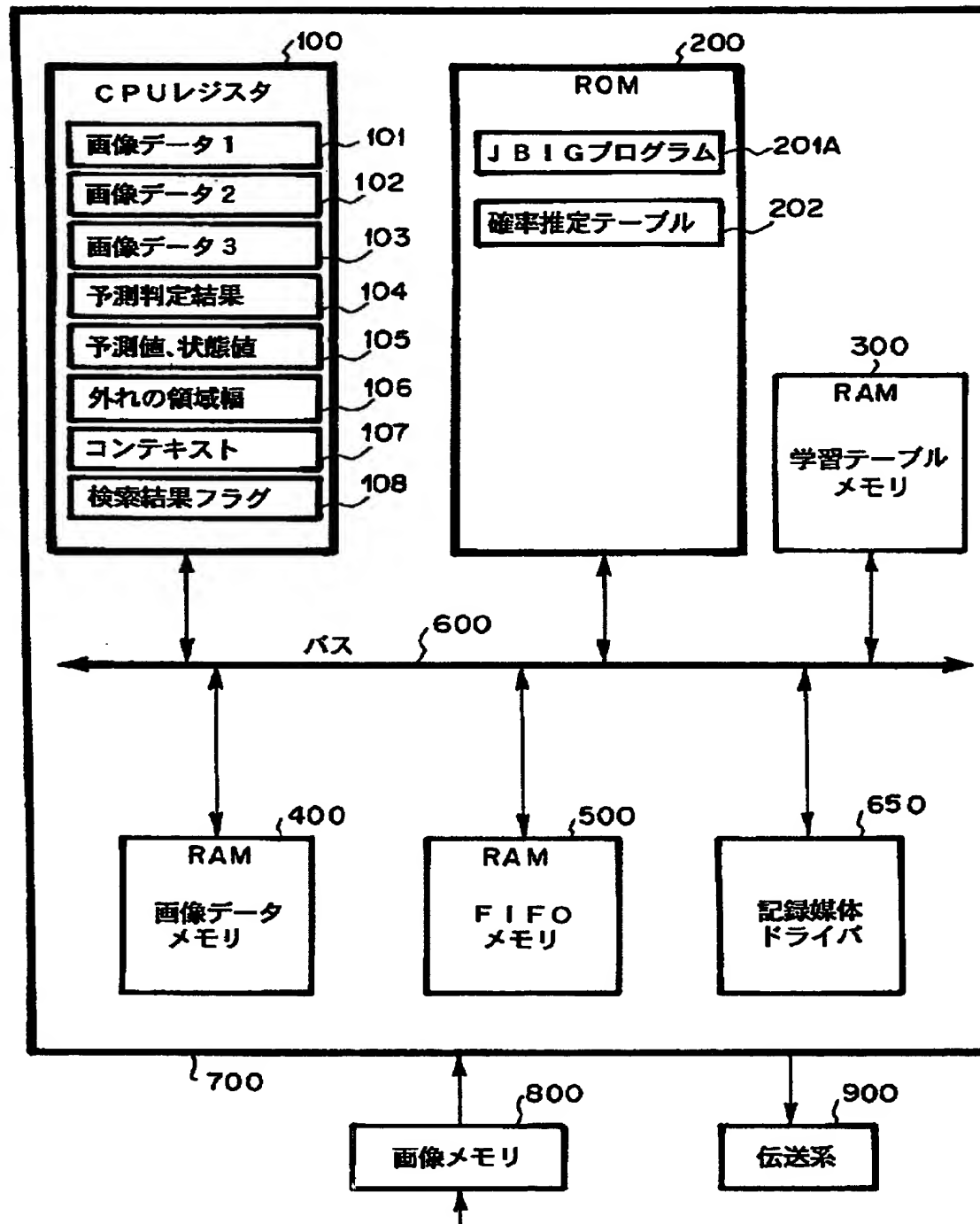
1 0 0 C P U

1 0 1, 1 0 2, 1 0 3 画像データレジスタ

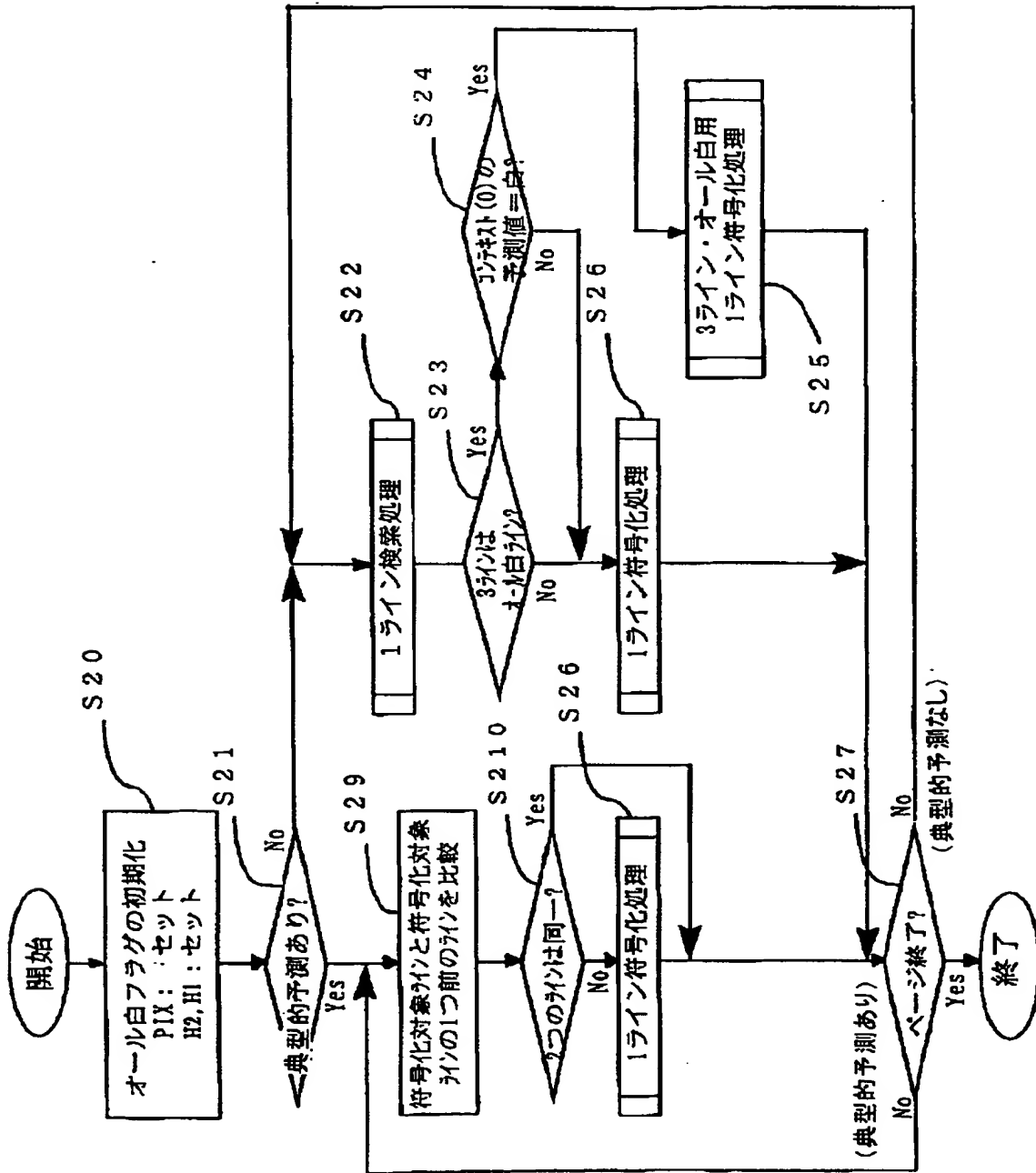
- 1 0 4 予測判定結果レジスタ
- 1 0 5 予測値、状態値レジスタ
- 1 0 6 外れの領域幅レジスタ
- 1 0 7 コンテキストレジスタ
- 1 0 8 検索結果フラグレジスタ
- 2 0 0 ROM
- 2 0 1 A J B I G プログラム
- 2 0 2 確率推定テーブル
- 3 0 0 RAM : 学習テーブル
- 4 0 0 RAM : 画像データメモリ
- 5 0 0 RAM : F I F O メモリ
- 6 5 0 記録媒体ドライバ
- 7 0 0 J B I G 方式符号化処理装置
- 8 0 0 画像メモリ
- 9 0 0 伝送系

【書類名】 図面

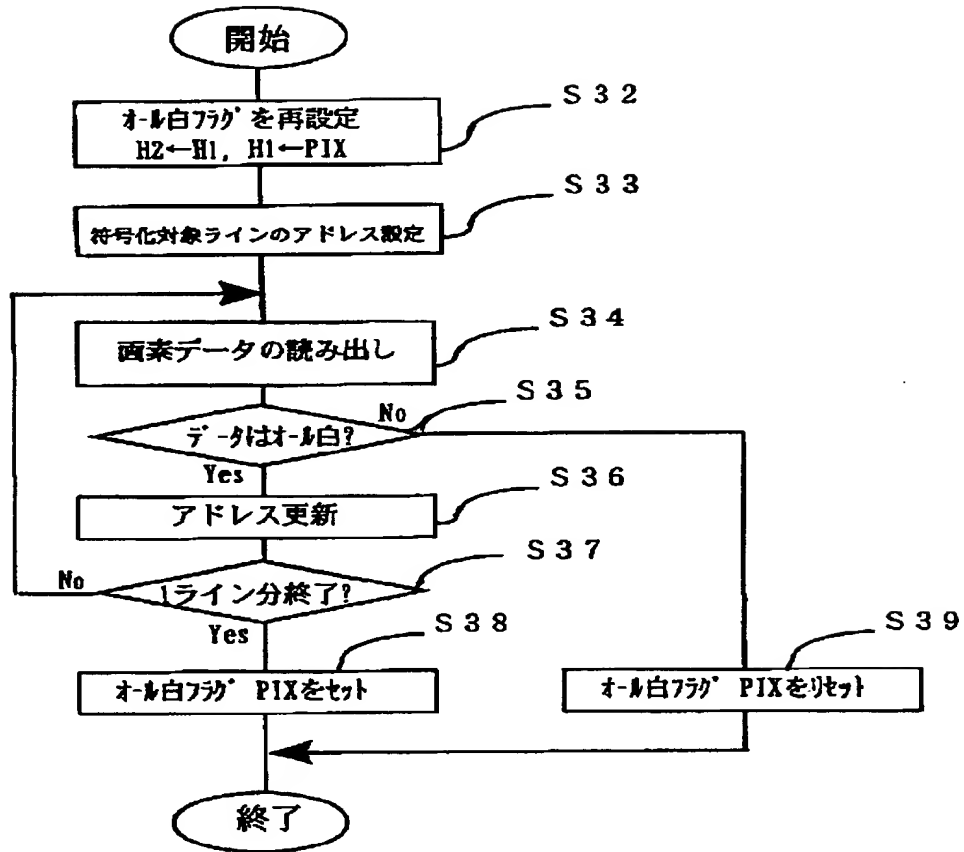
【図 1】



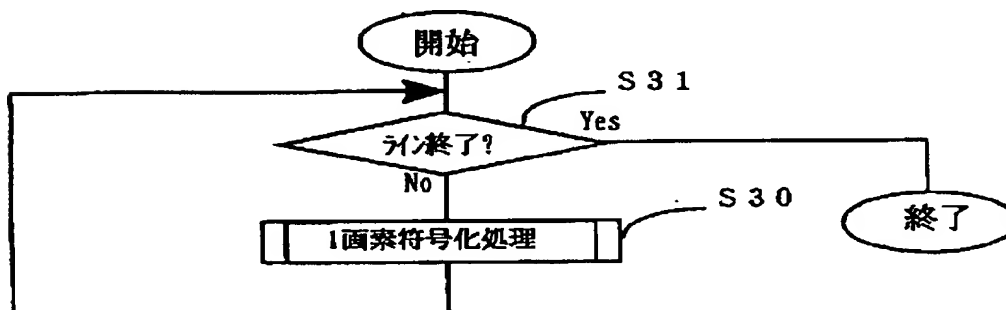
【図2】



【図 3】

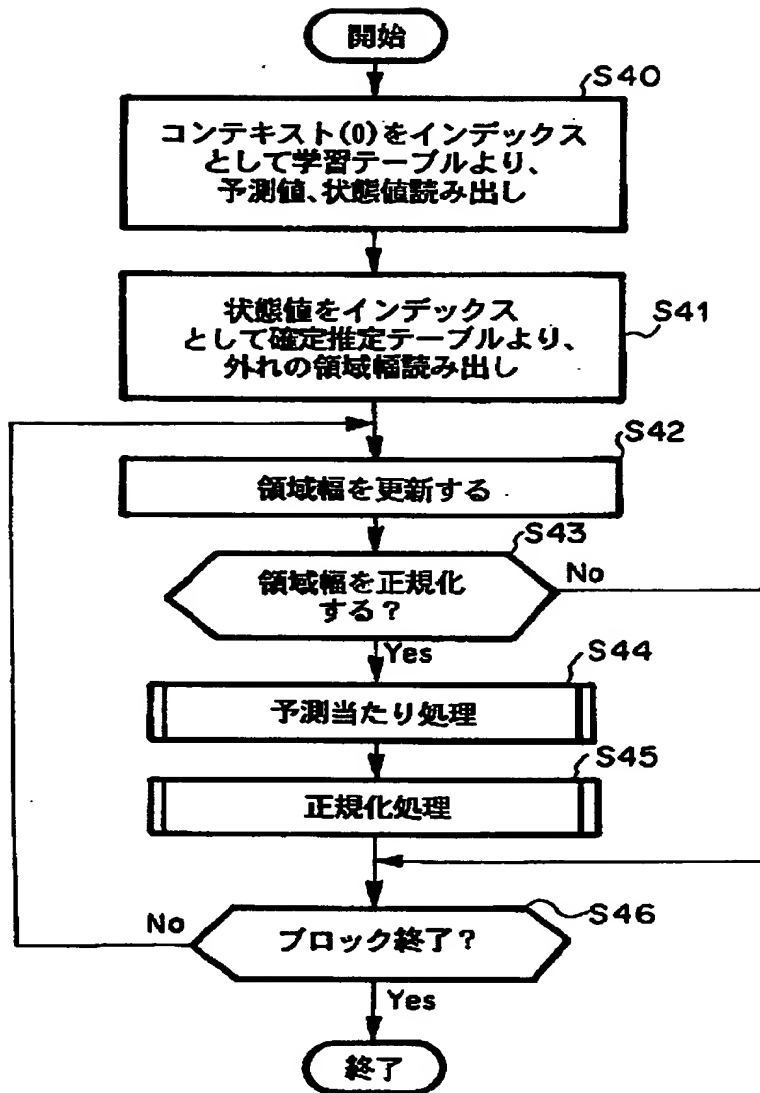


(a) 1ライン検索処理

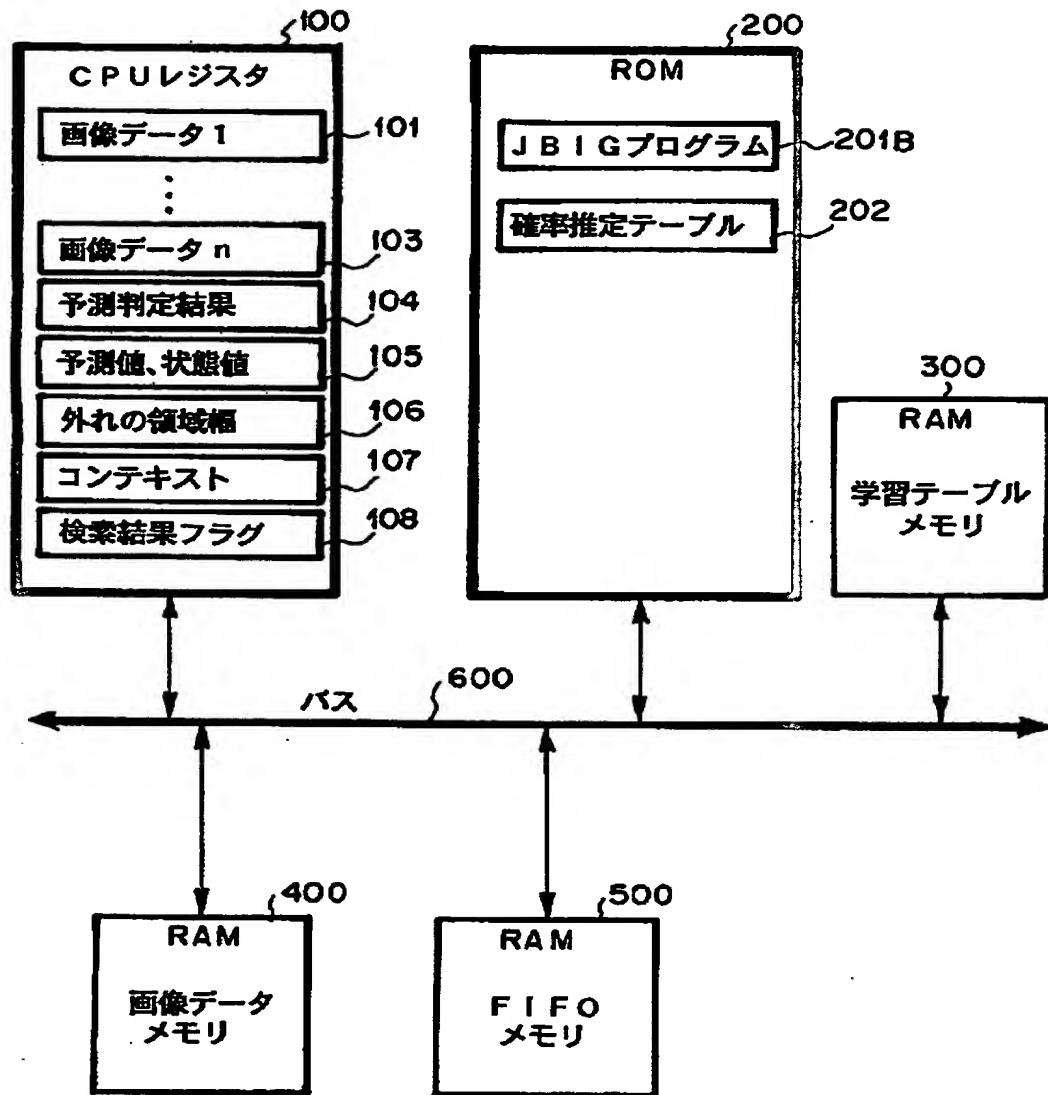


(b) 3ライン・オール白用1ライン符号化処理

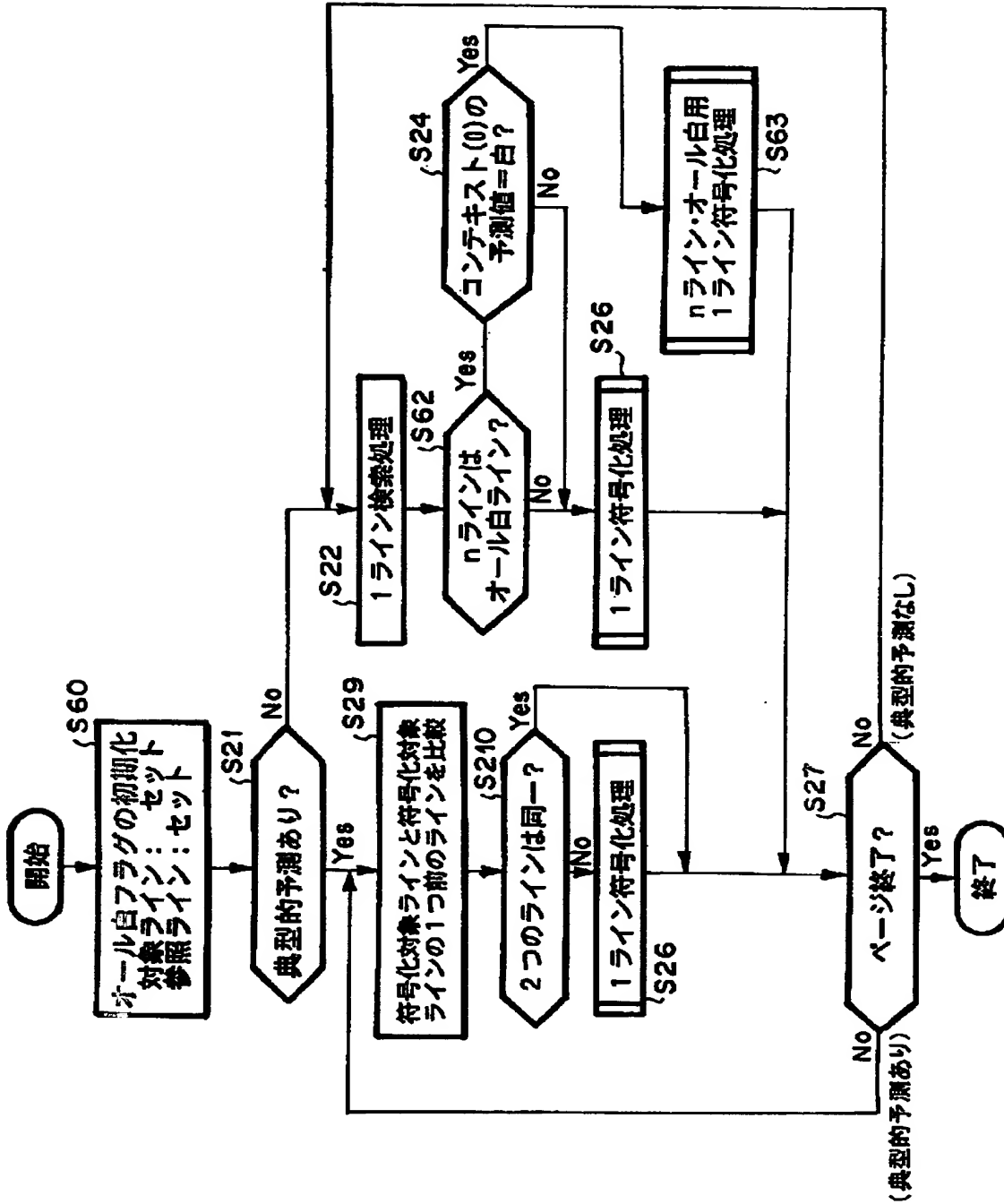
【図 4】



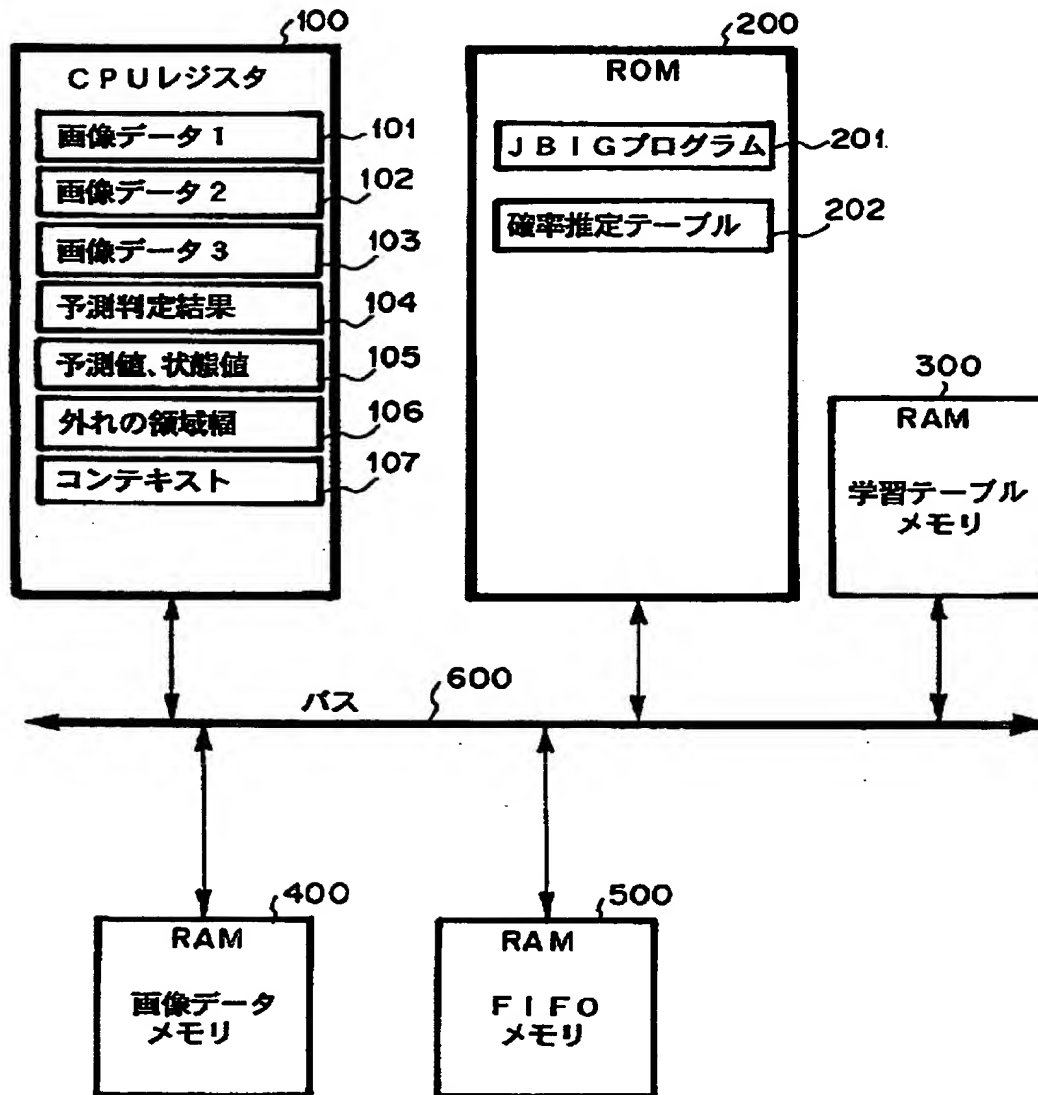
【図5】



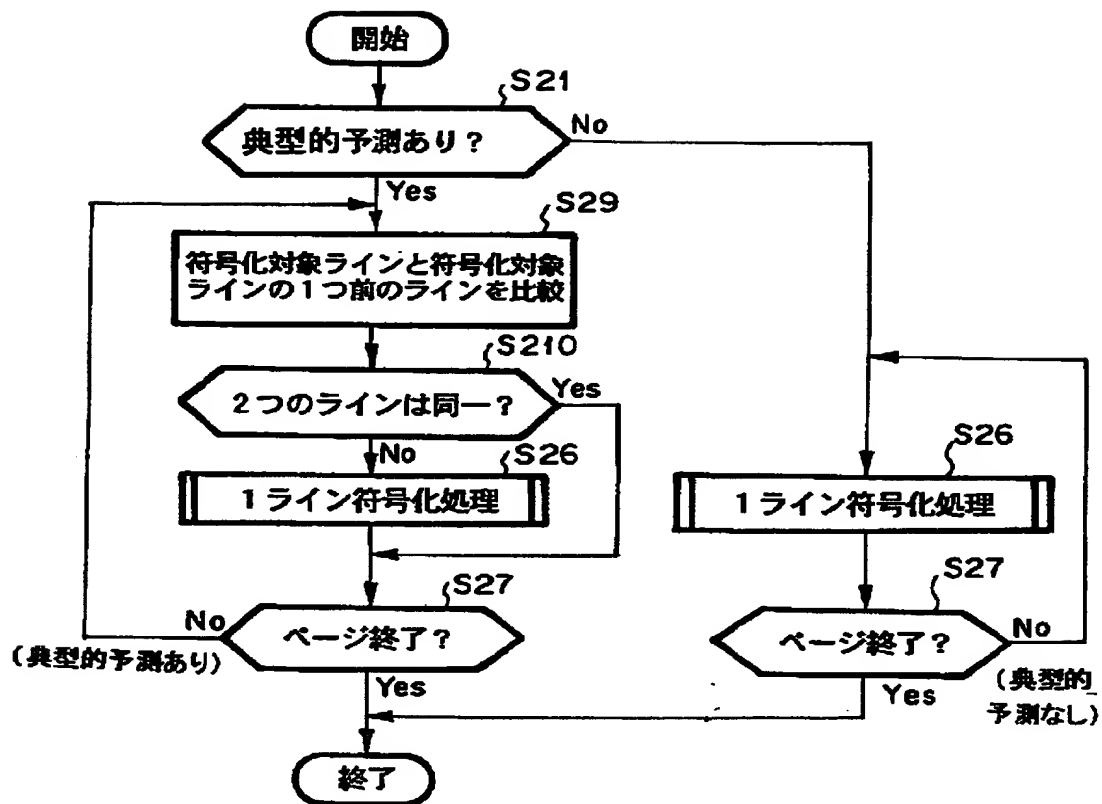
【図 6】



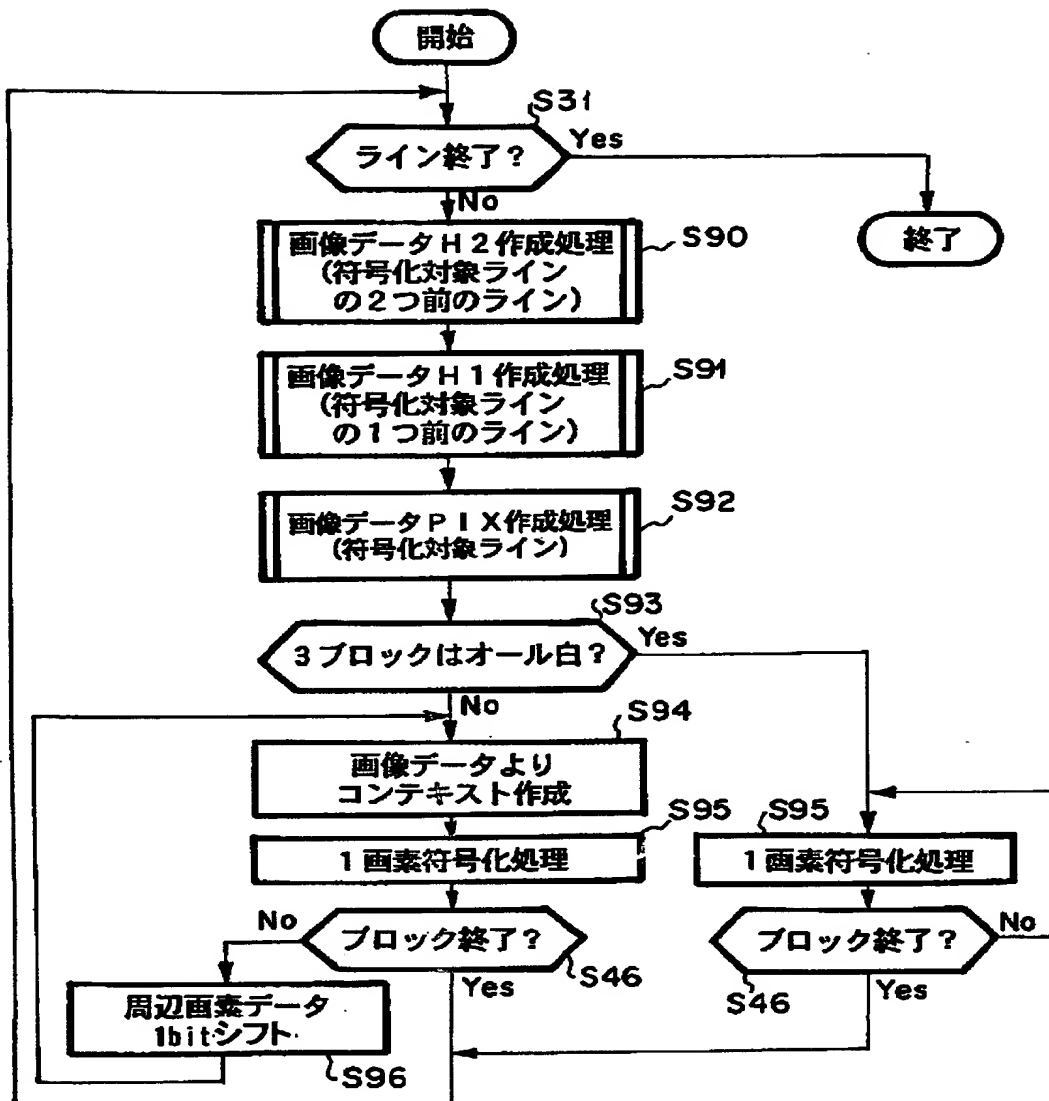
【図 7】



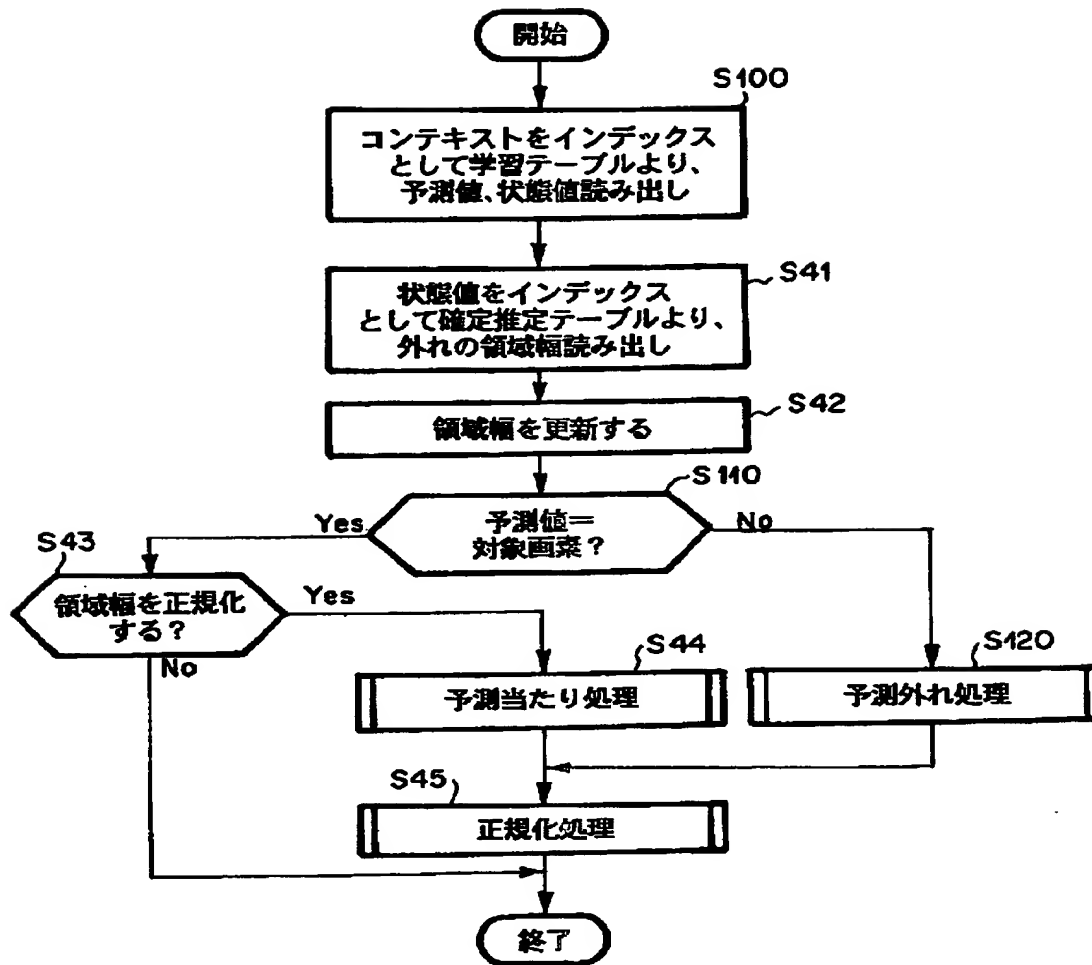
【図 8】



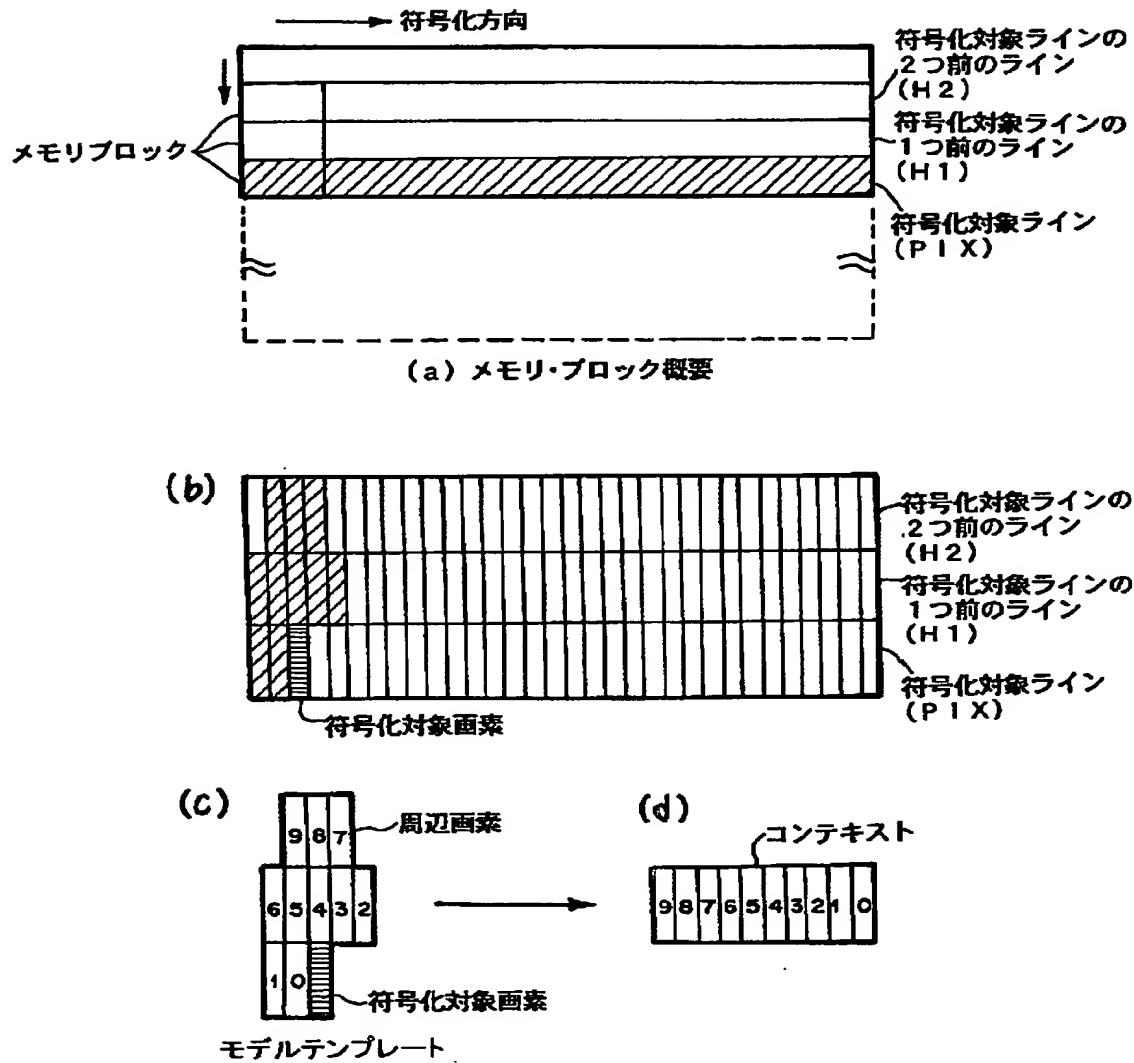
【図9】



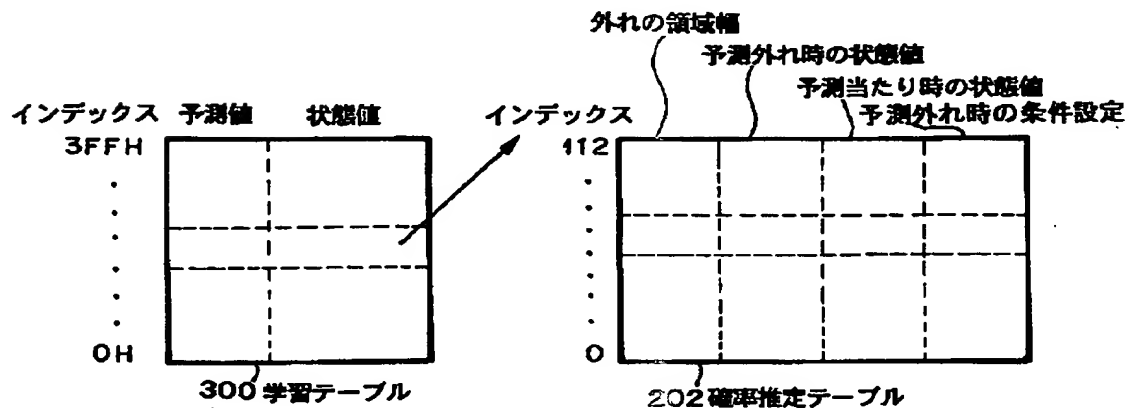
【図10】



【図 1 1】



【図 1 2】



【書類名】 要約書

【要約】

【課題】 画像データ圧縮方法或いは J B I G 方式符号化処理方法において、プログラム処理速度を高速化し、かつ用いる記録媒体の効率を向上することを課題とする。

【解決手段】 入力された画像データをライン毎にブロック単位で読み出し、そのブロック単位の画像データ毎に前記画像データの符号化対象画素の周辺画素を参照してコンテキストを作成して、2 値画素データに圧縮を行う画像データ圧縮方法において、前記コンテキスト作成処理は、典型的予測の有無を判断し、前記典型的予測のない場合に、まず、1 ライン分が白ラインか否かを検索し、続いて 3 ラインが全て白ラインか否かを判断し、オール白の場合前記コンテキストの予測値が白であるのか否かを判断し、前記コンテキストの予測値が白である場合に前記 3 ラインオール白用のライン専用の第 1 の 1 ライン符号化処理を実行し、前記 3 ラインが全て白ラインでない場合、及び前記コンテキストの予測値が白でない場合に第 2 の 1 ライン符号化処理を実行することを特徴とする。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 2 3 2 0 3 6]

1. 変更年月日	1 9 9 0 年 8 月 1 3 日
[変更理由]	新規登録
住 所	神奈川県川崎市中原区小杉町 1 丁目 4 0 3 番 5 3
氏 名	日本電気アイシーマイコンシステム株式会社